Machulskis Sergey

sergeym@tuta.io
Russia, Novosibirsk, Akademgorodok

Experience

- since 2019 – Positive Technologies. Senior programmer.
  - Built parts of a threat intelligence platform. Integrated third-party threat intelligence feeds.
  - Fixed a huge chunk of tech debt. Increased write/read throughput by two orders of magnitude.
  - Helped the team to release the platform on tight schedule.
- 2017 - 2019 – Novosibirsk State University. Assistant professor.
  - Taught courses on operating systems (system calls), parallel programming (pthreads) and computing history.
- 2015 - 2019 – 2GIS. Senior programmer.
  - Maintained a CDN delivering map data to 20 millions of users. Automated operator workflow so updated maps could be delivered in one click daily. The updates were monthly before that. Improved Python code quality and performance a lot. The service could churn a task for a week, the run time was brought to two hours.
  - Built a fault-tolerant cross datacenter pipeline based on `Apache Kafka`. It replaced an old single datacenter `ZeroMQ`-based pipeline. Made the transition seamless for consumers.
  - Brought a new life to a dying Python-based geopositioning service. It was slow, hard to scale, and fault intolerant. Rewrote and stabilized it using `asyncio`. As the load increased with years, rewrote it in `Go`. Now it handles heavy load on commodity hardware.
  - Created and executed several long-term team roadmaps. Decomposed, estimated and prioritized hundreds of tasks. Actively participated in sprint planning and process improvement. Coordinated cross-team feature development.

Education

- 2009 – 2015. Novosibirsk State University. Master's degree in Information Technology. Specialized in parallel computing.

Languages

- *Python*. Has been using it almost every day for 5 years straight. Wrote a thousand tests using `pytest`. Type-hinted an entire codebase to increase maintainability and to find hidden bugs. Learned principles of good Python code from libraries like `aiohttp`, `SQLAlchemy` and `Requests`. Can explain and use `import this`. Contributed to `mypy`.
- *Go*. 4 years of experience in sum. The last year was devoted to Go exclusively. Wrote a dozen services from scratch, rewrote Python services in Go (it was very satisfying) and supported huge monolithic apps. Understands how to model complex domains and how to build good architecture using Go. Wrote a couple of blog posts about Go, for example, a guide to structured logging in Go and a note on database/sql connection pools.
- Also built applications using `C`, `C++`, `Java` and `Scala`, but this knowledge is probably outdated. Here's an example of C++ he used to write. Prefers not to use those languages for new projects.

Profile

- *Learning*. Lifelong learner. There's a bookshelf.
- *GNU/Linux*. Built his own Linux from scratch (at home). Knows enough about UNIX-like OSs to teach it at the university. Comfortable with DevOps work involving containers (`Docker`), orchestration (`Ansible`, blog post), monitoring (`Grafana`, `Zabbix`), CI (`Jenkins`, `Gitlab`) and logs (`syslog`, `Elastic`). Practices `12-factor` apps.
- *Distributed systems*. Defended a thesis on this topic. Appreciates the motto "Reliability, Maintainability, Scalability!". Can explain what's happening in `tcpdump` + `wireshark`, `Zookeeper` and `WAL`s.
- *Databases*. Used `Redis`, `Tarantool`, `PostgreSQL` and `ClickHouse`. Writes efficient SQL.
- *Documentation*. Knows how to document architecture, APIs, SLAs, postmortems, tutorials and so on.
- *UX*. Cares about UX even though it's "not backend responsibility".
- *Architecture*. Uses Clean architecture and domain-driven development in practice.
- *Communication*. Knows how to find real problems in generic complaints. Knows common cognitive biases. Understands that programming is not just about writing code. The code ends up being less important than the architecture, and the architecture ends up being less important than social issues.